Skyline Software Systems, Inc. v. Keyhole, Inc et al Doc. 16

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF MASSACHUSETTS

SKYLINE SOFTWARE SYSTEMS, INC., Plaintiff, v. KEYHOLE, INC., and GOOGLE INC. Defendants.	CIVIL ACTION NO. 06-10980 DPW

DECLARATION OF PROFESSOR STEVEN K. FEINER, Ph.D., IN SUPPORT OF DEFENDANTS' MOTIONS FOR SUMMARY JUDGMENT OF NONINFRINGEMENT AND ANTICIPATION

[PUBLIC REDACTED VERSION]

Nelson G. Apjohn (BBO No. 020373)
NUTTER McCLENNEN & FISH LLP
World Trade Center West
155 Seaport Boulevard
Boston, MA 02210
(617) 439-2000
Fax: (617) 310-9000

Attorneys for Defendants and Counterclaimants
KEYHOLE, INC. and GOOGLE INC.

Of Counsel
Darryl M. Woo, admitted *pro hac vice*
Michael J. Sacksteder, admitted *pro hac vice*
Heather N. Mewes, admitted *pro hac vice*
Carolyn Chang, admitted *pro hac vice*
Saundra L.M. Riley, admitted *pro hac vice*
FENWICK & WEST LLP
Embarcadero Center West
275 Battery Street
San Francisco, CA 94111
Tel. (415) 875-2300
Fax (415) 281-1350
email: dwoo@fenwick.com

Dockets.Justia.com

I, Steven K. Feiner, declare as follows:

1. I make this declaration in support of the motions for summary judgment of noninfringement and invalidity filed by Defendants Keyhole, Inc. and Google Inc. ("Google"). In this declaration, I provide my expert opinions concerning certain issues in this lawsuit relating to United States Patent No. 6,296,189 ("the '189 patent").

2. My qualifications are stated more fully in my *curriculum vitae,* a true and correct copy of which is attached as Exhibit A.

3. I received a Ph.D. in Computer Science from Brown University in 1987. I received an A.B. degree in music from Brown University in 1973.

4. I am presently a Professor of Computer Science at Columbia University, a position that I have held for twenty years. I have been a Full Professor since January 2000. Prior to that, I was an Associate Professor of Computer Science at Columbia University from January 1991 until December 1999, and an Assistant Professor from September 1985 to December 1990. Prior to joining the faculty of Columbia University in September 1985, I was a Research and Teaching Assistant in the Department of Computer Science at Brown University from September 1977 until August 1985.

5. At Columbia University, I direct the Columbia University Computer Graphics and User Interfaces Laboratory, and teach both graduate and undergraduate students in computer graphics and user interfaces courses. I advise Computer Science doctoral candidates, primarily in the field of computer graphics and user interfaces. I am an active academic researcher, whose areas of research include knowledge-based design of graphics and multimedia, user interfaces, virtual reality and augmented reality, wearable computing, animation, hypermedia, and visualization.

6. I am coauthor of *Computer Graphics: Principles and Practice, Second Edition,* Addison-Wesley, 1990 ("*Computer Graphics*"), an authoritative and frequently cited academic computer graphics text. I am also a coauthor of *Introduction to Computer Graphics*, Addison-Wesley, 1993, and *Computer Graphics: Principle and Practice, Second Edition in C*, Addison-Wesley, 1996. As indicated on my *curriculum vitae*, I am the author and coauthor of over thirty journal papers, over seventy conference papers, and numerous other workshop papers, books and book chapters, editorials and other publications on computer graphics and user interfaces. I have been an Associate Editor of *ACM Transactions on Graphics* and *ACM Transactions on Information Systems*, and have been on the editorial boards of *IEEE Transactions on Visualization and Computer Graphics*, and *Virtual Reality*. I am a frequent invited speaker on computer graphics and user interfaces at institutions such as Princeton University, the Massachusetts Institute of Technology, and Carnegie Mellon University. In addition, I have given invited talks at numerous conferences and workshops, including ones related to Geographic Information Systems ("GIS"), such as *GIScience 2002*, the Advanced Research and Development Activity *Geospatial Intelligence Information Visualization Researchers Meeting 2003* and *GIS Planet 2005*. In 1991, I received an Office of Naval Research Young Investigator Award.

7. I am a named inventor on an issued United States patent relating to computer graphics, entitled "Worlds-within-worlds nested display and interaction system and method" (U.S. Pat. No. 5,524,187).

8. I have previously submitted declarations in this case in support of Google's responsive claim construction brief and in support of Google's opposition to Skyline's motion for preliminary injunction. I have also submitted declarations in support of Google's Opening

Claim Construction Brief and Responsive Claim Construction Brief filed for the November 1, 2006 Claim Construction Hearing.

9. I have reviewed the '189 patent and its relevant prosecution history and am familiar with this patent, its claims, and the background technology.

10. The '189 patent uses concepts, nomenclature, designs, and systems from the computer graphics art that should be understood in this context. In my opinion, one of ordinary skill in the art relevant to the subject matter of the '189 patent at the time the application for the patent was filed would be a person with a bachelor's degree in Computer Science, including at least one course in computer graphics, or with academic or work experience equivalent to that level of education.

11. I understand that Skyline Software Systems, Inc. asserts that certain claims of the '189 patent are infringed by certain Google products, including Keyhole 2LT, Keyhole 2 PRO, Keyhole 2 EC, Keyhole 2 NV, Keyhole 2 FUSION LT, Keyhole FUSION PRO and server, Google Earth, Google Earth Plus, Google Earth Pro, Google Earth Enterprise Client, Google Earth Server, Google Earth Fusion, and the Geo Coder Server (collectively, the "accused Google Earth products"). *See* Declaration of Carolyn Chang in Support of Defendants' Motions for Summary Judgment ("Chang Decl."), Ex. 3.

12. I have provided and summarized my expert opinions that the accused Google Earth products do not infringe the asserted claims of the '189 patent in my August 10, 2006 Rebuttal Expert Report and my December 22, 2006 Rebuttal Expert Report. A copy of my August 10, 2006 Rebuttal Expert Report, without the accompanying exhibits, is attached to this Declaration as Exhibit B. A copy of my December 22, 2006 Rebuttal Expert Report, without the accompanying exhibits, is attached to this Declaration as Exhibit C.

13. In addition, I have provided and summarized my expert opinions that the asserted claims of the '189 patent are invalid in my December 8, 2006 Expert Report. A copy of my December 8, 2006 Expert Report, without the accompanying exhibits, is attached to this Declaration as Exhibit D.

I. ANALYSIS OF GOOGLE EARTH

A. All Asserted Claims: "downloading … if the provided block from the local memory is not at the indicated resolution level"

14. Each of the asserted claims of the '189 patent requires downloading one or more additional higher resolution data blocks "if the provided block from the local memory is not at the indicated resolution level." '189 patent, claim 1 at col. 16:38-44, claim 3 at col. 16:62-66, claim 7 at col. 17:53-57, claim 12 at col. 18:26-30, claim 14 at col. 18:66-19:3, and claim 18 at col. 20:19-21. The Court construed this term to mean "downloading … upon some determination that the block provided from local memory is not at the indicated resolution level." Chang Decl., Ex. 5 at 10-12, 27.

15. In providing this construction, the Court understood the asserted claims of the '189 patent to require "determination as to whether *the conditional* is satisfied before actions predicated on it are taken." *Id.* at 11 (emphasis added). Thus, the Court recognizes that the downloading of additional data blocks claimed in the '189 patent is predicated on the condition that the first block provided from the local memory is not at the indicated resolution level. *Id.* [redacted] Therefore, in my opinion, the Google Earth client does not meet this limitation of the asserted claims of the '189 patent.

16. Skyline's expert claims that [redacted]

[redacted] I have reviewed the portions of the code identified by Skyline's expert, and in my opinion they do not meet the "downloading ... if the provided block from the local memory is not at the indicated resolution level" limitation.

17. [redacted]

18. Thus, in the Google Earth client [redacted]

19. [redacted]

[redacted] In the asserted claims of the '189 patent, the conditional that must be satisfied before downloading from a remote server is that the block from local memory is not at the indicated resolution level. Chang Decl., Ex. 5 at 11. [redacted]

20. Indeed, in Google Earth, [redacted], which fails to satisfy the "block from local memory is *not at* the indicated resolution level" conditional that is claimed in the '189 patent. For example, Google Earth [redacted] Therefore, the Google Earth client does not meet the "downloading ... upon some determination that the block provided from local memory is not at the indicated resolution level" limitation.

B. All Asserted Claims: "receiving from the renderer one or more coordinates in the terrain along with indication of a respective resolution level"

21. To provide data blocks describing three-dimensional terrain to a renderer, each of the asserted method and apparatus claims of the '189 patent require "receiving from the renderer one or more coordinates in the terrain along with indication of a respective resolution level." '189 patent, claim 1 at col. 16:32-34, claim 3 at col. 16:55-57 ("a plurality of coordinates in the terrain"), claim 7 at col. 17:46-48, claim 12 at col. 18:21-23, claim 14 at col. 18:61-63, and claim 18 at col. 20:14-16.

22. Thus, the asserted claims of the '189 patent all require a "renderer." The Court construed "renderer" as it is used in the '189 patent as a "software and/or hardware object that performs at least the following functions: (1) determining and providing to another object the required coordinates in the terrain along with a respective resolution level; (2) receiving the data blocks corresponding to the specified coordinates; and (3) using the received data blocks to display a three-dimensional image." Chang Decl., Ex. 4 at 26-32, 39.

23. Pursuant to the Court's construction, the claimed "renderer" must be a software and/or hardware object. *Id.* at 26-32, 39. The Court explained that an "object" does not necessarily have to be a software object of the sort used in object-oriented programming. *Id.*, Ex. 5 at 9 n.7. However, the Court's construction does require the "renderer" to be something that can be considered a logical entity or thing. *Id.*

24. The Court construed the "receiving from the renderer" limitation as requiring "something distinct from the renderer" to receive the required coordinates and respective resolution level from the renderer. Chang Decl., Ex. 5 at 8-10, 27. The Google Earth client also does not meet this limitation.

25. Skyline's expert claims [redacted]

[redacted]

26. [redacted]

27. [redacted] Accordingly, it is my opinion that Google Earth does not meet the "receiving from the renderer one or more coordinates in the terrain along with indication of a respective resolution level" limitation of the asserted claims of the '189 patent.

28. In addition, the parts of the Google Earth source code identified by Skyline's expert as purportedly meeting the "renderer" limitation are logically disjoint, and would not be considered a logical entity or thing. [redacted]

[redacted]. *See* Chang Decl., Ex. 6.

29. In my opinion, these distinct parts of the Google Earth source code do not form a logical object or thing as required by the "renderer" limitation. The '189 patent describes a system in which the "renderer" is a distinct logical entity that requests and receives specific data blocks by coordinates and a respective resolution level. '189 patent, Fig. 5 and col. 11:19-38. The Court understood this in construing "renderer" as a "software and/or hardware object" that determines and provides to "another object the required coordinates in the terrain along with a respective resolution level." Chang Decl., Ex. 4 at 32, 39. Skyline's expert simply identifies disjoint pieces of code that purportedly perform the stated functions of the '189 patent "renderer," but ignores the requirement that the functions must be performed by a single object or thing. Skyline's expert does not provide an explanation of how these separate source code functions can be considered a logical entity or thing.

C. All Asserted Claims: "data blocks belonging to a hierarchical structure"

30. Each of the asserted claims of the '189 patent discloses a method or apparatus for providing data blocks describing three-dimensional terrain to a renderer, the "data blocks belonging to a hierarchical structure." '189 patent, *e.g.*, claim 1 at col. 16:29-30, claim 3 at col. 16:52-53, claim 7 at col. 17:43-44, claim 12 at col. 18:13-14, claim 14 at col. 18:53-54, and claim 18 at col. 20:5-6.

31. The Court has construed "data blocks belonging to a hierarchical structure" to mean "data blocks that are organized into multiple levels of resolution, whereby each level contains data blocks at the same resolution, and each successive level contains data blocks of a higher resolution than those in the preceding level." Chang Decl., Ex. 4 at 12-15, 37.

32. Skyline's expert recognizes that, on occasion, [redacted]

33. In addition, [redacted]

34. [redacted] Google Earth does not meet the "data blocks belonging to a hierarchical structure" limitation.

D. All Asserted Claims: "data blocks describing three-dimensional terrain"

35. Each of the asserted claims of the '189 patent discloses a method or apparatus for

providing "data blocks describing three-dimensional terrain to a renderer." '189 patent, *e.g.*, claim 1 at col. 16:28-29, claim 3 at col. 16:51-52, claim 7 at col. 17:42-43, claim 12 at col. 18:12-13, claim 14 at col. 18:52-53, and claim 18 at col. 20:4-5.

36. The Court has construed "data block describing three-dimensional terrain" as "a block or collection of data or digital information that represents or describes a section of three-dimensional terrain at a particular resolution level and that includes any additional data overlaid on the digital image of the terrain, such as altitude, labels or optional objects." Chang Decl., Ex. 4 at 9-12, 37. The Court further construed "terrain" as "the surface features of an area of land, an object, or a material, including color, elevation, and existing objects or structures on the land, object or material." *Id.* at 17-19, 38.

37. In his August 11, 2006 Rebuttal Expert Report on the issue of the invalidity of the '189 patent, Skyline's expert interprets the Court's construction of "data blocks describing three-dimensional terrain" such that an imagery data file or a terrain data file alone would not satisfy this limitation. *See id.*, Ex. 8 at ¶ 23. Specifically, Skyline's expert states, [redacted]

38. In Google Earth, [redacted]

[redacted]

Therefore, if, the asserted claims of the '189 patent require both imagery and elevation (terrain) data to be included in a block or collection for it to constitute a "data block describing three dimensional terrain," then the Google Earth client does not meet this limitation.

E. Claims 3 and 14: "wherein blocks of lower resolution levels are downloaded before blocks of higher resolution levels"

39. In addition to the claim limitations already discussed, claims 3 and 14 contain a further limitation of "downloading from a remote server one or more additional data blocks … ***wherein blocks of lower resolution levels are downloaded before blocks of higher resolution levels***." '189 patent, claim 3 at col. 16:62-67 and claim 14 at col. 18:66-19:3. In my opinion, Google Earth also does not infringe claims 3 and 14 because it does not meet the "wherein blocks of lower resolution levels are downloaded before blocks of higher resolution levels" limitation.

40. Claims 3 and 14 of the '189 patent disclose a specific download order: downloading lower resolution blocks before higher resolution blocks. The Court construed "downloading" as "requesting over a network from a separate computer and receiving on a local computer." Chang Decl., Ex. 5 at 4-8, 27. To meet this limitation of claims 3 and 14, data blocks of lower resolution must be requested and received before data blocks of higher resolution are requested and received.

41. [redacted]

[redacted]

42. [redacted]

However, claims 3 and 14 disclose a specific download order with lower resolution data *required* to be requested and received before higher resolution data. [redacted]

[redacted] Even under a random download ordering scheme, lower resolution data will on occasion be requested and received before higher resolution data. In my opinion, such chance ordering of the downloading is insufficient to meet the "wherein blocks of lower resolution levels are downloaded before blocks of higher resolution levels" limitation of claims 3 and 14.

F. Claims 7 and 18 (and their Dependent Claims): "when not downloading blocks required by the renderer"

43. Method claim 7 and apparatus claim 18 contain the further limitation of "downloading from a remote server excess data blocks not currently need by the renderer to fill up the local memory ***when not downloading blocks required by the renderer***." '189 patent, claim 7 at col 17:58-61 and claim 18 at col. 20:22-24 (emphasis added). Method claims 8-9 and 23, and apparatus claims 19, 22, and 24 depend from claims 7 and 18, and therefore also contain this limitation. *Id.* at claims 8-9, 19, and 22-24.

44. The Court construed "when not downloading blocks required by the renderer" to mean "during periods of time when the local computer, or a connection thereof, is not downloading data blocks in response to coordinates received from the renderer." Chang Decl., Ex. 5 at 17-20, 27.

45. Skyline's expert [redacted] However, this fails to address whether the downloading is performed "when not downloading blocks required by the renderer" as that term was construed by the Court (i.e., when "not downloading blocks in response to coordinates received from the renderer"). In construing this term, the Court recognized that "files which are, or were, 'required by the renderer' may include data blocks not required for the currently displayed view." *Id.*, Ex. 5 at 19.

46. Skyline's expert also suggests that the limitation is met in Google Earth when [redacted] However, the Court explicitly recognized that this is not downloading excess blocks "when not downloading blocks required by the renderer," stating that because "the '189 patent allows for a constantly changing user viewpoint, there will often be a queue of data blocks, some of which may not be needed any longer, that have been requested by the renderer, but not yet downloaded. Until these requested blocks have finished downloading, excess blocks will not be downloaded to fill the local memory. *See* Fig. 8 ('Queue Empty?' No --> Download blocks from queue; Yes --> Download excess surrounding blocks). Therefore, files which are, or were, 'required by the renderer' may include data blocks not required for the currently displayed view." *Id.*, Ex. 5 at 19 (footnote omitted).

47. Thus, under the Court's construction, the download of excess data blocks "when not downloading blocks required by the renderer" must take place after all requested blocks on the download queue have finished downloading. Skyline's expert argues that on a computer with a single network connection, because excess data blocks cannot be downloaded when blocks required by the renderer are being transmitted, the excess blocks are necessarily downloaded "when not downloading blocks required by the renderer." *Id.*, Ex. 9 at 18. [redacted].

48. In any event, any suggestion that the download of excess data blocks must by definition take place "when not downloading blocks required by the renderer" would render that limitation meaningless and redundant, improperly eliminating the limitation entirely.

49. Thus, in my opinion, upon a proper reading of the Court's construction of "when not downloading blocks required by the renderer," Google Earth does not meet this limitation.

II. ANALYSIS OF TERRAVISION

50. TerraVision was a system for visualization of terrain. It allowed a user to view, in real time, a synthetic recreation of a real landscape created from elevation data and a large number of aerial or satellite images of that landscape. I first learned of TerraVision at SIGGRAPH '95 in Los Angeles, before this lawsuit was filed. I recall seeing TerraVision being demonstrated in the Interactive Communities venue at SIGGRAPH '95. In addition, I received a copy of the Multimedia CD-ROMs published in 1995 by SIGGRAPH for SIGGRAPH '95. I recognize G-T 0021 as a copy of these materials.

51. I have reviewed Exhibit 206 to the deposition of Stephen Lau, a CD containing source code for TerraVision. *See* Chang Decl., Ex. 14 at 216:6-224:25, and Exs. 20, 34. In particular, I have reviewed the revision history for the source code, as reflected in the code itself, and this version of the source code dates from no later than April 27, 1996. In fact, a number of the files were last revised even earlier. There is no indication that any of the source code itself was modified or otherwise altered after April 27, 1996.

52. Below, I briefly describe certain relevant functions of the TerraVision source code. For convenience, I have described each function according to the file in which it is located.

53. TerraVision.c [src\TerraVision\TerraVision] includes the following functions:

a. **TerraVisionSpawnProcs**: This function spawns various threads, including the tile visibility thread, the rendering thread, and the tile requesting thread.

b. **TerraVisionInitDataSet**: This function initializes tile data structures and calls TSMSpawnThreads. It then downloads the DEM tiles by calling TsRequestDems. (Alternatively, if the "-local" flag is passed to TerraVision, this function will read the DEM tiles locally by calling TsReadDems.)

c. **TerraVisionRender**: This function executes as the rendering thread.

54. TsTsm.c [src\TerraVision\libTileSvc] includes the following functions:

a. **TSMSpawnThreads**: This function spawns a tile receiver thread for each server.

55. Visible.c [src\TerraVision\TerraVision] includes the following functions:

a. **GenerateVisible**: This function executes as the tile visibility thread. It loops, iteratively calling ThreeDWidgetGenerateVisible to determine how to generate the next frame.

b. **GenerateRequests**: This function calls ThreeDWidgetGenerateRequests to generate a quadtree. GenerateRequests then creates a list of the tiles in that quadtree in sorted order, from coarsest to finest resolution.

c. **TsMakeRequest**: This function executes as the tile requesting thread. It determines the rate at which requests to download tiles should be generated and loops, iteratively calling GenerateAndSendRequests.

d. **GenerateAndSendRequests**: This function waits for an alarm based on the download request update rate and then calls GenerateRequests to create a sorted list of the desired tiles (tiles up to the appropriate resolution) from coarsest to finest resolution. It then loops through this list in order, either placing a tile on a download request list if it is not already resident in the cache (by calling tsmReqTile), or updating the usage time on the tile if it is already resident (by calling TileMgrUpdateTile). If any tiles were placed on the download request list, then tsmStopReqTiles is called.

e. **ParseQuadTree**: This function takes a quadtree and returns a list of the "leaf" tiles that are resident in memory. It is called from ThreeDWidgetGenerateVisible.

56. ThreeDWidget1.c [src\TerraVision\TerraVision] includes the following functions:

a. **ThreeDWidgetGenerateVisible**: This function calls ThreeDWidgetCalcVisibility to generate a quadtree of visible tiles at up to the appropriate resolution based on the user's current view matrix. It then calls ParseQuadTree to determine the "leaf" tiles in the quadtree that are resident in memory and calls ThreeDWidgetCreateRenderPrimitive to create the mesh that will be rendered during the next frame.

b. **ThreeDWidgetGenerateRequests**: This function also calls ThreeDWidgetCalcVisibility to generate a quadtree of requested tiles at up to the appropriate resolution, based on a "bloated" view matrix (i.e., a view matrix specifying an expanded version of the view frustum used by ThreeDWidgetGenerateVisible).

c. **ThreeDWidgetCalcVisibility**: This recursive function creates a quadtree whose nodes are contained within the frustum defined by the passed view matrix up to the appropriate resolution based on that view matrix.

d. **ThreeDWidgetCreateRenderPrimitive**: The function creates the underlying polygonal mesh that will be used to draw the scene, based on the view matrix and the visible tiles.

57. tsmTileIO.c [src\tsmApi] includes the following functions:

a. **tsmReqTile**: This function places on a download request list a record for a tile containing specified coordinates at a specified resolution.

b. **tsmStopReqTiles**: This function is called after all calls to tsmReqTile have been made for a frame. It calls a function specific to the kind of protocol that will be used to download the tiles, including tsmStopReqTiles_iss and tsmStopReqTiles_web.

c. **tsmGetNextTileHeader**: This function gets the next tile's header by calling a function specific to the kind of protocol being used to download the tile, including tsmGetNextTileHeader_iss and tsmGetNextTileHeader_web.

d. **tsmGetNextTileData**: This function gets the next tile's data by calling a function specific to the kind of protocol being used to download the tile, including tsmGetNextTileData_iss and tsmGetNextTileData_web.

58. tsmTileIO_iss.c [src\tsmApi] includes the following functions:

a. **tsmStopReqTiles_iss**: This function calls lower-level functions that send the tile requests to a remote server.

b. **tsmGetNextTileHeader_iss**: This function calls a lower-level function to read a tile's header from a remote server.

c. **tsmGetNextTileData_iss**: This function calls a lower-level function to read a tile's data from a remote server.

59. tsmTileIO_web.c [src\tsmApi] includes the following functions:

a. **tsmGetTile_web**: This function calls tsmHttpUrlToBuffer to get the tile through HTTP.

b. **tsmStopReqTiles_web**: This function does very little.

c. **tsmGetNextTileHeader_web**: This function does very little.

d. **tsmGetNextTileData_web**: This function calls tsmGetTile_web.

60. tsmUrlToFile.c [src\tsmApi] includes the following functions:

a. **tsmHttpUrlToBuffer**: This function reads all data at a remote server's URL into a passed buffer.

61. TsTileStruct.c [src\TerraVision\libTileSvc] includes the following functions:

a. **TsImageServerReader**: This function executes as the tile receiver thread. It loops, reading tiles by calling TsReadTileFromTSM.

b. **TsReadTileFromTSM**: This function reads a tile by calling tsmGetNextTileHeader and tsmGetNextTileData.

c. **TsReadDems**: This function reads the DEM tiles from files in the file system.

d. **TsRequestDems**: This function calls TsRequestDEMLevel for each level of the hierarchy and then calls tsmStopReqTiles.

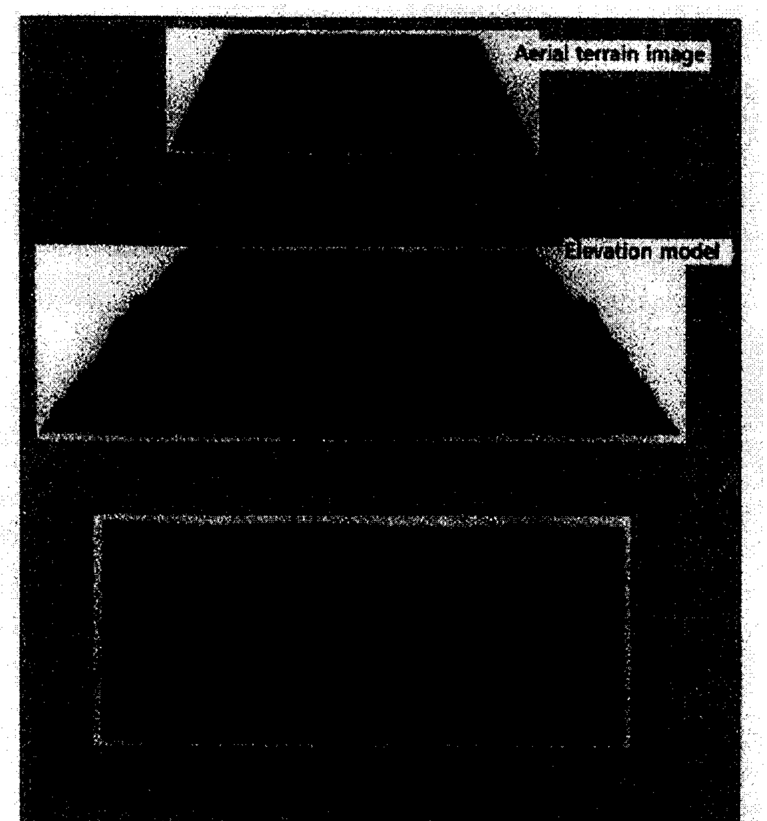
e. **TsRequestDEMLevel**: This function calls tsmReqTile for each DEM tile at one level of the hierarchy.

A. All Asserted Claims: "providing data blocks describing three-dimensional terrain"

62. The asserted claims recite a method of or apparatus for "providing data blocks describing three-dimensional terrain data." The Court has construed "data block describing three-dimensional terrain" to mean "a block or collection of data or digital information that represents or describes a section of three-dimensional terrain at a particular resolution level and that includes any additional data overlaid on the digital image of the terrain, such as altitude, labels or optional objects." Chang Decl., Ex. 4 at 9-12.

63. The TerraVision application disclosed "providing data blocks describing three-dimensional terrain." TerraVision was a high-speed graphics application that allowed a user to interact in real time with a synthetic 3D photo-realistic view of a large terrain. This application accessed "tiles" representing both elevation and image data. Tiles with elevation data were referred to as digital elevation model ("DEM") tiles and tiles with image data were referred to as

orthographic image ("OI") tiles. Both DEM tiles and OI tiles were data blocks describing three-dimensional terrain data as construed by the Court. They were data blocks or collections of digital information that represented or described a section of three-dimensional terrain, that is the "surface features of an area of land, an object, or a material, including color, elevation, and existing objects or structures on the land, object or material." Moreover, both DEM tiles and OI tiles were provided, as rendering of the terrain on the screen in TerraVision was accomplished by combining the DEM and OI tiles for the selected area at the appropriate resolution, as illustrated in Figure 4 of the MAGIC IEEE Article:



■ Figure 4. *Mapping an ortho-image onto its digital elevation model. (Source: SRI International)*

Chang Decl., Ex. 22 at GOOG 350.

B. All Asserted Claims: "renderer"

64. The asserted claims further indicate that the "data blocks describing three-

dimensional terrain" are provided to a "renderer." The Court has construed the "renderer" to be a "software and/or hardware object that performs at least the following functions: (1) determining and providing to another object the required coordinates in the terrain along with a respective resolution level; (2) receiving the data blocks corresponding to the specified coordinates; and (3) using the received data blocks to display a three-dimensional image." *Id.*, Ex. 4 at 26-32.

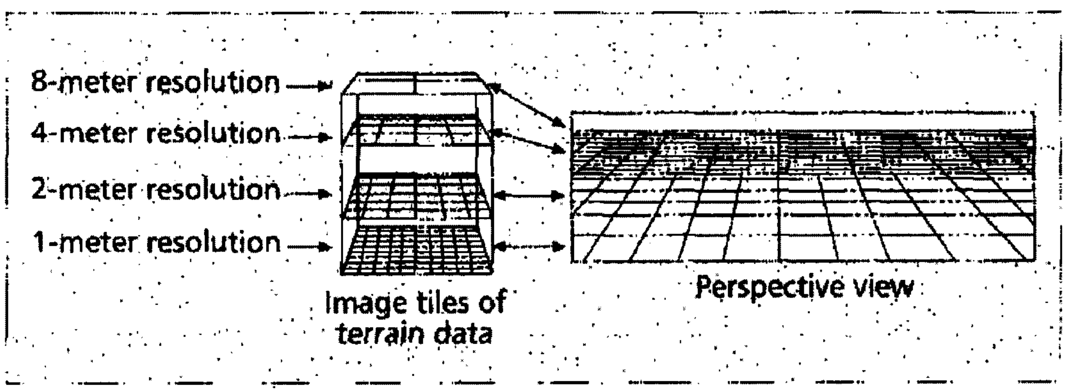
65. In my opinion, the TerraVision application disclosed a "renderer" as construed by the Court. The TerraVision source code confirms that the TerraVision application had software that performed the functions of (1) determining and providing to another object the required coordinates in the terrain along with a respective resolution level; (2) receiving the data blocks corresponding to the specified coordinates; and (3) using the received data blocks to display a three-dimensional image. Specifically, ThreeDWidgetGenerateVisible calls ThreeDWidgetCalcVisibility to create a quadtree identifying all the visible tiles within the frustum up to a resolution appropriate for the view matrix. Each tile is identified by an *x* coordinate, a *y* coordinate and a resolution level (e.g., QuadTile). ParseQuadTree (another object) receives the quadtree from ThreeDWidgetGenerateVisible (part of the renderer). ParseQuadTree then provides a list of the "leaf" tiles resident in memory to ThreeDWidgetGenerateVisible, which uses these tiles to create the underlying polygonal mesh that will be used to display the scene. Finally, the rendering thread will actually display the three-dimensional image based on the tiles provided.

C. All Asserted Claims: "data blocks belonging to a hierarchical structure"

66. The asserted claims further recite "data blocks belonging to a hierarchical structure which includes blocks at a plurality of different resolution levels." The Court has

construed "data blocks belonging to a hierarchical structure" as "data blocks that are organized into multiple levels of resolution, whereby each level contains data blocks at the same resolution, and each successive level contains data blocks of a higher resolution than those in the preceding level." Chang Decl., Ex. 4 at 12-15.

67. The TerraVision application disclosed "data blocks belonging to a hierarchical structure which includes blocks at a plurality of different resolution levels," as that phrase has been construed by the Court. TerraVision used a multiresolution hierarchy or pyramid of increasingly lower-resolution representations of the equal-sized tiles. Each level in the pyramid was at half the resolution of the previous level. This is illustrated in Figure 3 of the MAGIC IEEE Article:



■ Figure 3. *Relationship between tile resolutions and perspective view. (Source: SRI International)*

Id., Ex. 22 at GOOG 349.

68. The TerraVision source code also confirms that TerraVision had "data blocks belonging to a hierarchical structure which includes blocks at a plurality of different resolution levels," as that phrase has been construed by the Court. The ThreeDWidgetCalcVisibility function in the TerraVision source code creates a quadtree identifying tiles at a plurality of different resolution levels. It can do this because the tiles on the remote servers have been

organized as pyramids. Thus, the quadtree created by ThreeDWidgetCalcVisibility and the pyramids on the remote servers are organized into multiple levels of resolution, where each level has tiles at the same resolution, and each successive level has tiles of a higher resolution than those in the preceding level.

D. All Asserted Claims: "receiving from the renderer one or more coordinates in the terrain along with indication of a respective resolution level"

69. The Court has construed "coordinates in the terrain" as "a set of numerical values that identifies a particular location in the terrain," and "terrain" as "the surface features of an area of land, an object, or a material, including color, elevation, and existing objects or structures on the land, object or material." Chang Decl., Ex. 4 at 17-23. The Court has also clarified that "receiving from the renderer" means "something distinct from the renderer receiving from the renderer." *Id.*, Ex. 5 at 8-10.

70. The TerraVision application disclosed the step of "receiving from the renderer one or more coordinates in the terrain along with indication of a respective resolution level." As discussed above, ThreeDWidgetGenerateVisible calls ThreeDWidgetCalcVisibility to create a quadtree with all of the visible tiles within the frustum up to a resolution appropriate for the view matrix. Each tile is identified by an *x* coordinate, a *y* coordinate and a resolution level (QuadTile). ParseQuadTree (another object) receives this quadtree from ThreeDWidgetGenerateVisible.

71. Claim 3 includes further language in the "receiving" step referring to the "plurality of coordinates being included in a plurality of respective distinct blocks." The Court has construed this language to mean "several coordinates, where each set of coordinates is contained within one block in a set composed of data blocks that are distinct from one another." Chang Decl., Ex. 5 at 14-16.

72. The Court's construction appears to encompass two different situations. First, where the renderer calls for a set of blocks using the (*x*,*y*) coordinates and resolution level of each block. And second, where the renderer calls for a set of blocks using coordinates of the boundaries of the necessary areas and a resolution level. *See* '189 patent, col. 14:10-16. Thus the coordinates may refer to one block or they may refer to many blocks. Chang Decl., Ex. 5 at 16 ("I construe this language more broadly, allowing a one-to-many relationship between the blocks and the coordinates"). As discussed above with respect to the other claims, the renderer in TerraVision calls for a set of blocks using the (*x*,*y*) coordinates and resolution level of each block, and thus falls within the Court's construction of claim 3 as well.

E. All Asserted Claims: "providing the renderer with a first data block which includes data corresponding to the one or more coordinates, from a local memory"

73. The Court has construed "first data block" as "the first data block provided to the renderer from the local memory corresponding to the specified coordinates." *Id.*, Ex. 4 at 15-17. The Court has construed "local memory" as "memory easily accessible to the user's processor, either because it is physically part of the processor or is attached directly thereto, and distinct from the memory of the remote server from which data must be downloaded." *Id.* at 32-34. The Court has also clarified that "providing the renderer" means "something distinct from the renderer providing to the renderer." *Id.*, Ex. 5 at 8-10.

74. The TerraVision application disclosed "providing the renderer with a first data block which includes data corresponding to the one or more coordinates, from a local memory." ThreeDWidgetGenerateVisible calls ParseQuadTree, which identifies a "first data block which includes data corresponding to the one or more coordinates, from a local memory." Specifically, ParseQuadTree (something distinct from the renderer) identifies the visible tiles resident in memory and provides them to the renderer.

75. There are some minor variations in the language of the "providing" step of claim 3. However, these differences do not change my analysis: the TerraVision application practiced this step for the same reasons discussed above with respect to the other asserted claims.

F. All Asserted Claims: "downloading from a remote server one or more additional data blocks at a resolution level higher than the resolution level of the first data block which include data corresponding to the one or more coordinates if the provided block from the local memory is not at the indicated resolution level"

76. The Court has construed "downloading" to mean "requesting over a network from a separate computer and receiving on a local computer." Chang Decl., Ex. 5 at 4-8. The Court has construed "downloading … if the provided data block from the local memory is not at the indicated resolution level" to mean "downloading … upon some determination that the block provided from local memory is not at the indicated resolution level." *Id.* at 10-12.

77. The TerraVision application disclosed "downloading from a remote server one or more additional data blocks at a resolution level higher than the resolution level of the first data block which include data corresponding to the one or more coordinates if the provided block from the local memory is not at the indicated resolution level," as construed by the Court. ThreeDWidgetGenerateRequests calls ThreeDWidgetCalcVisibility to generate a quadtree of requested tiles at up to the appropriate resolution based on a "bloated" view matrix (i.e., a view matrix specifying an expanded version of the view frustum used by ThreeDWidgetGenerateVisible). This quadtree is passed to GenerateAndSendRequests as a sorted list, in coarse-to-fine resolution order. The tiles in the list include those in local memory, as well as those on a remote server. GenerateAndSendRequests then loops through the list, in coarse-to-fine order, to determine whether or not each tile is in local memory. If a tile in the list is already in local memory, then GenerateAndSendRequests will cause the usage time on the tile to be updated, so as to prevent the tile from being deleted from local memory, and it will not

attempt to download that tile from a remote server. When GenerateAndSendRequests encounters a tile that is not in local memory (i.e., as it continues its coarse-to-fine traversal), then it will place that tile on a download request list. The download request list is processed by the tile set manager, which actually downloads the tiles from a remote server. The tile set manager supports several alternative methods for downloading tiles, including the use of ISS protocols and the use of HTTP to access tiles addressed as URLs. GenerateAndSendRequests will request that additional tiles at a resolution level higher than the resolution level of the first tile provided from local memory be downloaded *if* the provided tile from local memory is not at the indicated resolution level. TerraVision thus downloaded additional tiles from the remote server "upon some determination that the block provided from local memory is not at the indicated resolution level."

78. In the TerraVision application, both OI tiles and DEM tiles were requested and downloaded from the remote server. DEM tiles were downloaded from the remote server when the user selected a data set, and then stored in local cache memory since there was only a relatively small number of DEM tiles. In fact, DEM tiles were requested in coarse-to-fine order in TsRequestDems, and the download list was created and processed using the exact same tsmReqTile and tsmStopReqTiles functions used for OI tiles. OI tiles were not downloaded when the user selected an initial data set, but rather would be downloaded only if the provided block in local memory was not at the indicated resolution level. OI tiles (like DEM tiles) were "data blocks [describing three-dimensional terrain]," and the downloading of these additional OI tiles satisfies the claim. The OI tiles described the appearance of the terrain and were needed (along with the DEM tiles) to render the three-dimensional terrain in the TerraVision application. *See also* ¶¶ 62-63 above.

G. Claims 3 and 14: "blocks of lower resolution levels are downloaded before blocks of higher resolution levels"

79. Claims 3 and 14 also includes further language in the "downloading" step that "blocks of lower resolution levels are downloaded before blocks of higher resolution levels." The TerraVision application requested blocks in a coarse-to-fine order (i.e., blocks of lower resolution levels are downloaded before blocks of higher resolution levels). In particular, GenerateAndSendRequests makes calls to tsmReqTile in coarse-to-fine order to generate a tile download request list that is sorted in order of increasing resolution level. The TerraVision application also received blocks in a coarse-to-fine order at least when it used the HTTP protocol to access tiles addressed as URLs. In TerraVision, this interface was implemented synchronously, meaning that TerraVision would wait to receive a requested block before requesting the next block. This is shown, for example, by tsmGetTile_web and tsmHttpUrlToBuffer.

H. Claims 7 and 18: "downloading from a remote server excess blocks not currently needed by the renderer to fill up the local memory when not downloading blocks required by the renderer"

80. Claims 7 and 18 include further language in the "downloading" step requiring "downloading from a remote server excess blocks not currently needed by the renderer to fill up the local memory when not downloading blocks required by the renderer." The Court has construed the phrase "when not downloading blocks required by the renderer" to mean "during periods of time when the local computer, or a connection thereof, is not downloading data blocks in response to coordinates received from the renderer." Chang Decl., Ex. 5 at 17-20

81. In the TerraVision application, the tile requesting and tile receiver threads downloaded the visible data blocks, as well as excess blocks not currently needed by the renderer to fill up the local memory. ThreeDWidgetGenerateRequests generates a quadtree of requested

tiles at up to the appropriate resolution, based on a "bloated" view matrix (i.e., a view matrix specifying an expanded version of the view frustum used by the tile visibility thread). This quadtree will include all the visible tiles identified by ThreeDWidgetGenerateVisible (i.e., the tiles needed by the renderer), as well as excess tiles from the bloated view matrix. GenerateAndSendRequests places both the visible tiles and the excess tiles on a download request list if they are not already in memory. The tile requesting and tile receiver threads will actually download these tiles from the remote server. The TerraVision application, however, did not first download the tiles needed by the renderer and then download the excess data blocks—these blocks were all placed on the same download request list in coarse-to-fine order. To the extent that Skyline seeks to construe claims 7 and 18 to not require giving priority to data blocks needed by the renderer, however, it is my opinion that under this broader construction, the TerraVision application would also disclose this limitation of claims 7 and 18.

82. The TerraVision Technical Note includes further disclosures. It provides that "[u]sing an expanded field of view of the predicted viewpoint is a relatively simple mechanism that has the advantage that exactly the same code used for traversing the terrain quad tree for the current view can be used to create a truncated quad tree for the future view." Chang Decl., Ex. 23 at GOOG 388. It further describes a "secondary list of 'tiles to pre-fetch if there's time'…." *Id.* In my opinion, at least the TerraVision Technical Note disclosed "downloading from a remote server excess blocks not currently needed by the renderer to fill up the local memory when not downloading blocks required by the renderer," as properly construed. The excess tiles are the tiles placed on the secondary list of tiles to pre-fetch if there's time. These tiles are only downloaded during periods of time when the local computer, or a connection thereof, is not downloading data blocks required by the renderer (i.e., "if there's time").

I. Claims 8 and 22: "downloading the blocks via the Internet" or "the communication link comprises a connection to the Internet"

83. Claims 8 and 22 include all the limitations of claims 7 and 18, and a further limitation relating to "downloading the blocks via the Internet" or having a communication link with "a connection to the Internet." The Court has construed the "Internet" to mean "[t]he publicly accessible network capable of relaying information via Internet Protocol, either alone or in conjunction with one or more other protocols, but not including a wholly self-contained private network of devices communicating only with each other." Chang Decl., Ex. 5 at 20-22. The Court further noted that "this construction is not intended to exclude networks such as Inte[r]net2, which are built on separate physical infrastructure, but are essentially updated and experimental versions of the current Internet." *Id.* at 22.

84. In my opinion, the TerraVision application downloaded blocks via the Internet. In addition to all of the evidence stating that TerraVision was in fact used over the Internet and not just the MAGIC network, the TerraVision source code includes functions for accessing tiles over the Internet by using the HTTP protocol, including, for example, tsmGetTile_web and tsmHttpUrlToBuffer.

85. Furthermore, in my opinion, the MAGIC network is akin to the Internet2 (i.e., it was an updated and experimental version of the then-current Internet). The MAGIC network was specifically designed as a testbed network. Furthermore, the MAGIC network was an ATM network. At the relevant time, ATM networks were gaining increasing acceptance as a high-speed Internet backbone. Moreover, even assuming that the MAGIC network is a private network, it was also part of the broader Internet, not a "self-contained group of computers or other devices communicating only with one another." Chang Decl., Ex. 5 at 21. For example, Stephen Lau testified that SRI was not on the MAGIC gigabit testbed, and therefore had to

access the MAGIC network over the Internet. *Id.*, Ex. 14 at 38:10–22, 281:25–283:1. This is confirmed in the MAGIC IEEE Article. *Id.*, Ex. 22 at GOOG 356 ("Proper testing of TerraVision and the ISS required high-speed interconnectivity. However, SRI and LBNL, the respective developers of these components, did not have such connectivity."). Accordingly, in my opinion, use of the MAGIC network over the broader Internet would also fall within the Court's construction.

I declare under penalty of perjury under the laws of the United States that the foregoing is true and correct. This declaration is executed this 19th day of January, 2007, in New York, New York.

Steven K. Feiner, Ph.D.

Certificate of Service

I hereby certify that, on January 19, 2007, I caused a true and accurate copy of the foregoing document to be served upon all counsel of record for each party by complying with this Court's Administrative Procedures for Electronic Case Filing.

By: /s/ Darryl M. Woo
Darryl M. Woo